(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **03231333 A**

(43) Date of publication of application: **15.10.91**

(51) Int. Cl **G06F 9/45**
G05B 15/02
G06F 9/06

(21) Application number: **02026129**

(22) Date of filing: **07.02.90**

(71) Applicant: **TOSHIBA CORP**

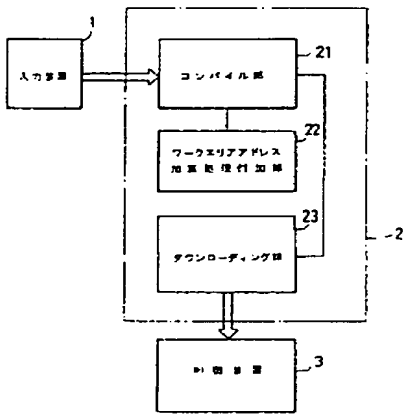
(72) Inventor: **WATANABE YASUE**

(54) **PROGRAM PRODUCING DEVICE AND CONTROLLER**

(57) Abstract:

PURPOSE: To attain the effective use of a memory by down-loading the size of an object program and the size of a working area used by the object program into this program.

CONSTITUTION: When the program data are inputted from an input device 1 via a programmer, a program production part 2 sends an input program to a compiling part 21. In the compiling state of the part 21, a working area address addition processing adding part 22 applies the working area address addition processing to the final part of a package using the working area. A down-loading part 23 down-loads the size of an object program and the size of the working area used by the object program into the program where the working area address addition processing is applied to the final part of the package compiled by the part 21 via a controller 3. Thus, the effective application is secured with the memory of the controller 3.



⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平3－231333

⑤1Int.Cl.⁵	識別記号	庁内整理番号	⑬公開　平成3年(1991)10月15日
G 06 F 9/45			
G 05 B 15/02		P 7740－5H	
G 06 F 9/06	430	D 7927－5B	
		8724－5B G 06 F 9/44 322 A	

審査請求　未請求　請求項の数 3　（全9頁）

⑤4発明の名称　プログラム作成装置および制御装置

⑳特　願　平2－26129
㉒出　願　平2(1990)2月7日

⑦2発明者　渡辺　恭江　東京都府中市東芝町1番地　株式会社東芝府中工場内
⑦1出願人　株式会社東芝　神奈川県川崎市幸区堀川町72番地
⑦4代理人　弁理士　鈴江　武彦　外3名

明　細　書

1．発明の名称

プログラム作成装置および制御装置

2．特許請求の範囲

（1）　入力されたプログラムをコンパイルするコンパイル部と、このコンパイル部のコンパイル時、ワークエリアを使用するパッケージの最終部にワークエリアアドレス加算処理を付加するワークエリアアドレス加算処理付加部と、前記コンパイル部によってコンパイルされた結果、パッケージ最終部にワークエリアアドレス加算処理が付加されたオブジェクトプログラムに、そのオブジェクトプログラムのサイズおよびオブジェクトプログラムで使用するワークエリアサイズを含めてダウンローディングするダウンローディング部とを備えたことを特徴とするプログラム作成装置。

（2）　前記請求項1に記載のプログラム作成装置からダウンローディングされたオブジェクトプログラムを記憶するプログラムメモリと、前記オブジェクトプログラムに含まれるパッケージ毎のワークエリアを記憶するワークエリアメモリと、前記オブジェクトプログラムを前記プログラムメモリの所定のアドレスに割り付けて記憶させるプログラムメモリ割付け部と、前記オブジェクトプログラムに含まれる各パッケージのワークエリアサイズに応じて前記ワークエリアメモリにパッケージ毎のワークエリアを割り付けるワークエリアメモリ割付け部と、前記プログラムメモリに記憶されているオブジェクトプログラムの実行管理を司るタスクスケジューラとを備えたことを特徴とする制御装置。

（3）　前記請求項第2項に、プログラメモリ中のオブジェクトプログラムから不使用になったプログラムおよびワークエリアを削除するパッキング部と、このパッキング部によるプログラムのパッキング時にプログラムメモリおよびワークエリアメモリにそれぞれプログラム毎のワークエリアを再割付けするメモリ再割付け手段とを付加してなることを特徴とする制御装置。

3．発明の詳細な説明

［発明の目的］

（産業上の利用分野）

本発明は、鉄鋼、紙パルプ、化学、食品プラントなどの一般産業設備や上下水道設備、廃棄物処理設備、ビル設備などのＦＡシステム等に適用するプログラム作成装置およびその作成プログラムを用いて制御対象を制御する制御装置に関する。

（従来の技術）

一般に、コンピュータ技術におけるワークエリアは、演算値のようなテンポラリデータを保存するレジスタ的な使用と、時系列的なデータの演算・参照を目的としてデータの保存を行う使用とが可能な２種類の使用形態をもったメモリエリアである。

そこで、従来、以上のような使用形態をもつワークエリアを有効に活用しつつプロセスの制御を実行するためにプログラム制御装置が広く利用されている。このプログラム制御装置で使用するプロセス制御プログラムのプログラミング言語としては、シーケンス制御用にはラダーダイヤグラムやＳＦＣ（Sequential Function Chart）、計装ループ制御用には計装ブロックなどのＰＯＬ言語が使用されている。

このＰＯＬ言語は制御に必要な最小単位の機能をパッケージ化（サブルーチン化）したものである。このサブルーチン化されたプログラムはおおむね制御タスク作成時には標準パッケージとして既に用意されているが、標準パッケージだけでは目的とする制御タスクを作成できない場合には、制御タスクを作成する前に要求仕様を満足するようなパッケージを新たに作成する必要がある。

この制御タスクは、用意されたパッケージ（サブルーチンプログラム）に、対象プロセス制御に適したパラメータを設定し、これを積み上げることにより作成することになる。

ところで、従来のワークエリア管理方式では、制御タスク作成時、その制御タスクのプログラマーがワークエリアを必要とするパッケージに対しワークエリアアドレスを人為的に固定割り付けするようにしている。このワークエリアの人為的な固定アドレス割付けは、制御タスク中のワークエリアを必要とするパッケージのワークエリアアドレス設定パラメータの端子にパッケージを使用するワークエリアアドレスを逐一記述しなければならないので、プログラム設計から保守に至るまでのエンジニアリングコストが大幅に増大する問題がある。

また、使用ワークエリア領域の予約・保守を人間系で行うことから不完全であり、ワークエリアの重複使用や記述ミスといった誤りも犯し易く、かつ、解析の困難なトラブルも発生する問題がある。

さらに、ワークエリアの固定アドレス割付けは、タスク単位での使用可能なワークエリアサイズが固定であるという制限にも関連しており、タスク単位でのワークエリアサイズの制限は制御タスク作成時にパッケージの組合わせによってはワークエリアが不足してくる場合もあり、本来の制御タスクのプログラミングを妨げるといった問題がある。

加えて、最近の制御装置では、プログラムメモリの有効活用を目的として、プログラムダウンローディング時のプログラムメモリ自動割付け処理や不要となった制御タスクを削除したり、制御タスクの入れ替えを行う場合のパッキング処理（使用不可の制御タスクプログラムのメモリに後続の制御タスクプログラムを前詰めに移動させることにより、メモリの不使用領域と使用領域との混在をなくす処理）を可能とする装置があるが、ワークエリアを固定アドレス割付けとした場合には制御タスクのダウンローディング後の削除時などに特にワークエリア領域内でメモリの不使用領域と使用領域とが混在し、プログラムメモリの有効活用の思想に合致しなくなる問題がある。

（発明が解決しようとする課題）

従って、以上述べたように従来のプログラム作成装置およびその制御装置は、制御タスクのワークエリアを人為的に固定アドレスにて割付けしているので、重複割付けや記述ミスによって解析

困難なトラブルが生じたり、ワークエリア不足によって効率的な制御タスクプログラミングが妨げられたり、さらにワークエリアの不使用領域および使用領域とが混在することによりプログラムメモリの有効活用が図れない問題がある。

本発明は上記実情に鑑みてなされたもので、制御装置単位でのプログラムワークエリアの自動割付けを行うことにより、エンジニアリングコストの削減、ワークエリアアドレスの設定ミスに起因するトラブルの撲滅、メモリの有効活用、プログラム作成／変更に柔軟に対処しうるプログラム作成装置およびその制御装置を提供することを目的とする。

〔発明の構成〕

（課題を解決するための手段）

先ず、請求項1の対応する発明は、入力されたプログラムをコンパイルするコンパイル部と、このコンパイル部のコンパイル時、ワークエリアを使用するパッケージの最終部にワークエリアアドレス加算処理を付加するワークエリアアドレス加算処理付加部と、前記コンパイル部によってコンパイルされた結果、パッケージ最終部にワークエリアアドレス加算処理が付加されたオブジェクトプログラムに、そのオブジェクトプログラムのサイズおよびオブジェクトプログラムで使用するワークエリアサイズを含めてダウンローディングするダウンローディング部とを備えたプログラム作成装置である。

次に、請求項2の対応する発明は、プログラム作成装置からダウンローディングされたオブジェクトプログラムを記憶するプログラムメモリと、前記オブジェクトプログラムに含まれるパッケージ毎のワークエリアを記憶するワークエリアメモリと、前記オブジェクトプログラムを前記プログラムメモリの所定のアドレスに割り付けて記憶させるプログラムメモリ割付け部と、前記オブジェクトプログラムに含まれる各パッケージのワークエリアサイズに応じて前記ワークエリアメモリにパッケージ毎のワークエリアを割り付けるワークエリアメモリ割付け部と、前記プログラムメモリに記憶されているオブジェクトプログラムの実行管理を司るタスクスケジューラとを備えた制御装置である。

さらに、請求項3に対応する発明は、請求項2の対応する発明に、パッキング処理機能を付加した構成である。

（作用）

従って、以上のような手段を講じたことにより、請求項1の発明は、入力装置を通して入力された制御タスクプログラムについてコンパイル部にてコンパイルを行ってオブジェクトプログラムを作成するとともに、タスクの総ワークエリアサイズを演算する。このとき、コンパイル対象がワークエリア使用パッケージの場合には、ワークエリアアドレス加算処理付加部がパッケージ最終部分にワークエリアアドレス加算処理を付加する。

しかる後、ダウンローディング部が制御装置に対してオブジェクトプログラムをダウンローディングする時には、オブジェクトプログラムのダウンローディングとともに、オブジェクトプログラム毎のワークエリアサイズもダウンローディングし、制御装置側でオブジェクトプログラム毎にそのワークエリアメモリ上に所定の大きさのワークエリアを自動的に割付けさせるようにするものである。

次に、請求項2の発明においては、プログラム作成装置からダウンローディングされたオブジェクトプログラムをプログラムメモリに記憶させるのであるが、そのときにプログラムメモリ割付け部がオブジェクトプログラムを所定のアドレスに自動的に割り付けて記憶させていく。また、ダウンローディングされてきたオブジェクトプログラム中で使用するワークエリアについてはワークエリアメモリ割付け部がワークエリアメモリの所定のアドレスに自動的に割り付けて行く。

このようにしてダウンローディングされたオブジェクトプログラムはタスクスケジューラにより起動され、制御のために使用される。

さらに、請求項3の発明は、パッキング部によりプログラメモリ中のオブジェクトプログラムか

ら不使用になったプログラムおよびワークエリアを削除し、同時にプログラムメモリ再割付け部により各オブジェクトプログラムをプログラムメモリ上の所定のアドレスに再割付けして記憶させ、またワークエリア再割付け部によりオブジェクトプログラム毎にそのワークエリアをワークエリアメモリ上に再割付けし、タスクスケジューラのプログラムの実行に供する。

（実施例）

以下、本発明の実施例について図面を参照して説明する。第１図は請求項１に係わるプログラム作成装置の一実施例を示すブロック図、第２図は請求項２に係わる制御装置の一実施例を示すブロック図である。

先ず、第１図に示すプログラム作成装置は、プログラマの操作に基づいてプログラムデータを入力する入力装置１のほか、この入力装置１から入力されたプログラムデータをコンパイルしてオブジェクトプログラムを作成するコンパイル部２１、このコンパイル部２１によってコンパイルされたオブジェクトプログラムからコールされる種々のパッケージにつき、ワークエリアを必要とするパッケージに対しそのパッケージ最終部にワークエリアアドレス加算処理を付加するワークエリアアドレス加算処理付加部２２、前記コンパイル部２１でコンパイルされたオブジェクトプログラムをそのタスクサイズ情報、ワークエリアサイズ情報とともにダウンローディングするダウンローディング部２３等を備えたプログラム作成部２が設けられている。このプログラム作成部２には同プログラム作成部２からダウンローディングされてくるオブジェクトプログラムを受けて所定の制御を実行する制御装置３が設けられている。

この制御装置３は、具体的には第２図に示すようにダウンローディングされてくるオブジェクトプログラムの入力処理を行う入力部３１と、この入力部３１から入力されるオブジェクトプログラムをプログラムメモリ３２に記憶するに際し、このプログラムメモリ３２にオブジェクトプログラム毎の先頭アドレスを自動的に割付けするプログラムメモリアドレス割付け部３３と、前記オブジェクトプログラムに含まれるパッケージ毎のワークエリアをワークエリアメモリ３４に記憶するに際しこのワークエリアメモリ３４にオブジェクトプログラム毎のワークエリアの先頭アドレスを自動的に割付けするワークエリアメモリ割付け部３５とが設けられている。

また、この制御装置３にはパッキング機能が設けられている。具体的には、プログラムメモリ中のオブジェクトプログラムから不使用になったプログラムおよびワークエリアを削除するパッキング部３６と、パッキング後のオブジェクトプログラム毎の先頭アドレスをプログラムメモリ３２に再割付けするプログラムメモリアドレス再割付け部３７と、同じくパッキング後のオブジェクトプログラム毎のワークエリアの先頭アドレスをワークエリアメモリ３４に再割付けするワークエリア再割付け部３８とで構成されている。

さらに、制御装置３には、プログラムを実行するためのオブジェクトプログラム先頭アドレス、ワークエリア先頭アドレス、タスクサイズ、ワークエリアサイズを登録して管理する管理テーブルメモリ３９と、この管理テーブルメモリ３９に対しそれらの必要な情報を記憶させるための管理テーブル作成部４０と、プログラム実行・管理を司るタスクスケジューラ４１と、このタスクスケジューラ４１によって取り出されたワークエリア先頭アドレスがセットされるワークエリアアドレスレジスタ４２，…とが設けられている。

次に、以上のように構成されたプログラム作成装置およびその制御装置の動作について説明する。プログラマによって入力装置１からプログラムデータが入力されると、プログラム作成部２では、その入力されたプログラムをコンパイル部２１にてコンパイルした後、ダウンローディング部２３に送り、ここでコンパイルされたタスクオブジェクトプログラムのほか、そのタスクオブジェクトプログラムサイズ情報およびタスクで使用するワークエリアサイズ情報等を制御装置３にダウンローディングする。

しかして、前記コンパイル部２１では第３図および第４図に示すようなコンパイル処理を実行する。先ず、第３図は制御タスクオブジェクトのコンパイル処理を示す図であって、制御タスクで使用するワークエリアサイズＷｓ(ti)を初期化，つまり零にクリアした後（ＳＴ１）、タスク終了命令があったか否かを判断し（ＳＴ２）、タスク終了命令があればコンパイル処理を終了し、タスク終了命令がなければパッケージのコール命令をセットする（ＳＴ３）。しかる後、コールするパッケージがワークエリアを使用するパッケージであるか否かを判断し（ＳＴ４）、ワークエリアを使用するパッケージであればタスク使用ワークエリアサイズＷｓ(ti)に該当パッケージで使用するワークエリアサイズＷｓ(pi)を加算する（ＳＴ５）。そして、以上の処理はタスクエンド命令に至るまでＳＴ３に戻って繰り返し行われる。なお、制御タスクコンパイル終了時のワークエリアサイズに保存された値が制御タスクで使用する総ワークエリアサイズとなる。

次に、第４図はコンパイル部２１におけるパッケージオブジェクトのコンパイル処理を示す図であって、パッケージ本来の機能処理部をコンパイルし（ＳＴ１１）、そのコンパイルされたパッケージがワークエリアを使用するパッケージであるか否かを判断する（ＳＴ１２）。ここで、ワークエリアを使用するパッケージであるならば、パッケージ最終部分においてワークエリアアドレスレジスタＲｘに、該パッケージで使用するワークエリアサイズを加算する処理を付加する（ＳＴ１３）。

一方、制御装置３では、プログラムメモリアドレス割付け部３３、ワークエリア割付け部３５によりタスクサイズ、ワークエリアサイズを基準にしてプログラムメモリ３２、ワークエリアメモリ３４の空きエリアに自動割付けをし、管理テーブル作成部４０により管理テーブルメモリ３９にタスク先頭アドレスＴａ、ワークエリア先頭アドレスＷａ、タスクサイズＴｓ、ワークエリアサイズＷｓを登録し、同時にプログラムメモリ３２のタスク先頭アドレスＴａからタスクオブジェクトプログラムを格納していく。

そして、このタスクの起動タイミングとなった場合、タスクの実行管理を司るタスクスケジューラ４１は、管理テーブルメモリ３９からワークエリア先頭アドレスＷａを取り出し、ワークエリアアドレスレジスタ４２，…にセットし、このタスクを起動して制御対象４を制御する。

次に、第５図は制御タスクプログラムメモリ３２を用いて制御タスクを構成するパッケージを呼び出す場合の処理系統図である。すなわち、同図に示すようにタスクオブジェクトプログラム５は、パッケージＳ１０，Ｓ２０，Ｓ２５，Ｓ３１をそれぞれ呼び出す「パッケージ呼び出し処理」とＥＮＤ命令とによって構成されている。この各々のパッケージのワークエリアサイズは、例えばパッケージＳ１０が６バイト、Ｓ２０が０バイト、Ｓ２５が２バイト、Ｓ３１が４バイトであるとすると、この制御タスクのワークエリアサイズＷｓは全体で１２バイトである。

なお、第５図では、タスクオブジェクト５とパッケージオブジェクト６とは別エリアに登録されているように示しているが、実際には同一のプログラムメモリ３２上にタスクオブジェクトとパッケージオブジェクトとが混在した形で格納されているものである。

タスクスケジューラ４１により起動された制御タスクは、先ず、「パッケージＳ１０呼び出し処理」を実行する。つまり、「パッケージＳ１０呼び出し処理」でコールされたパッケージオブジェクト６のパッケージＳ１０はその機能処理中でワークエリアアドレスレジスタで指定されたワークエリアアドレス（Ｗａ）から６バイト分のワークエリアのデータの読み出し／書き込みを行い、パッケージＳ１０の処理が終了したならば、ワークエリアアドレスレジスタＲｘに自己パッケージで使用するワークエリアサイズを加算して（Ｗａ＋６）番地をＲｘを保存し、制御タスクのパッケージコール命令の次番地にリターンする。

次は、「パッケージＳ２０の呼び出し処理」で

あるが、このパッケージはワークエリアを使用しないので、コールされたパッケージＳ２０本来の機能処理のみを実行する。

次の「パッケージＳ２５の処理」と「パッケージＳ３１の処理」とは前記「パッケージＳ１０処理」と同様にそれぞれのパッケージ本来の機能処理を終了後、ワークエリアアドレスレジスタＲｘに自己パッケージで使用するワークエリアサイズを加算し、次のパッケージで使用するワークエリア先頭アドレスとして、それぞれ（Ｗａ＋８）、（Ｗａ＋１２）を与える。このパッケージ処理の連続により制御タスクが実行される。なお、タスクの先頭から最後まで、ワークエリアアドレスレジスタＲｘは破壊されてはならない。

これら一連の処理の系列機能分担は、大別すると第６図に示すように人間系８、プログラム作成装置系９、制御装置系１０の３種類があり、以下、系別にその処理機能を説明する。

先ず、人間系８では、制御装置３内のワークエリア先頭アドレスおよびサイズを指定する。このサイズはタスク単位ではなく制御装置一括単位であるので、このサイズをオーバしない限り制御タスク単位でのワークエリアサイズは無制限に使用可能である。

引き続き、制御タスク構成要求であるパッケージを作成する。ここで、ワークエリアを使用するパッケージならば、コンパイルデータとして必要なのでパッケージのワークエリア使用サイズを登録する。但し、標準パッケージだけで制御タスクの作成が可能なシステムの場合には、パッケージの作成の必要はない。最後に、制御タスクの作成を行う。このとき、制御タスクを構成するパッケージのパラメータとしてワークエリアアドレスを指定する必要はない。

次に、プログラム作成装置系９は、前記人間系８において制御タスク作成の完了後にコンパイルを実行することにより行う。このとき、前述の第３図および第４図に示す処理に基づき、制御タスクコンパイル時には制御タスクのワークエリア総使用サイズが計算される。また、パッケージのコンパイル時にはワークエリアを使用するパッケージの最終部には、ワークエリアアドレス加算処理が付加される。

このコンパイルが完了した後、制御装置系１０に対して制御タスクのダウンローディングを行う。この時のダウンローディング項目は、タスクサイズ、ワークエリアサイズ、タスクオブジェクトである。

次に、制御装置系１０では、ダウンローディングデータを受信すると、第２図に示したようにプログラムメモリ３２とワークエリアメモリ３４とにプログラムパッケージの先頭アドレス、ワークエリア先頭アドレスの自動割り付けをを行い、制御タスクプログラムを実行する。

ダウンローディング処理や制御タスクの削除処理を繰り返すうちに、パッキング処理が必要になった場合には制御タスク実行の合間にパッキング部３６によりパッキング処理を行い、削除されていない制御タスクのプログラムメモリとワークエリアメモリとの内容を前詰めにパッキングし、新しいタスク先頭アドレスとワークエリア先頭アドレスとをそれぞれプログラムメモリアドレス再割付けをし、それらのアドレス情報を管理テーブル作成部４０により管理テーブルメモリ３９に登録する。

以上のようにして制御装置３のダウンローディング時、パッキング時において、常にタスクのワークエリア先頭アドレスを自動割付けし、タスク・サブルーチンを管理する管理テーブルにそれらのアドレス情報を保存し、タスクスケジューラ４１が制御タスクを起動するごとに管理テーブルよりタスクに割付けられたワークエリア先頭アドレスを取出し、ワークエリアアドレスレジスタＲｘにセットし、制御タスクプログラムを実行する。

そして、制御タスク（オブジェクトプログラム）は、プログラムロジックの順番にしたがってパッケージをコールする。ここで、コールされたパッケージがパッケージ本来の処理の中でワークエリアを使用するものであれば、ワークエリアアドレ

スレジスタＲｘにより指定されたワークエリア先頭アドレスから自己パッケージで使用するワークエリアサイズ分のエリアのデータの読み出し／書き込みを実行し、パッケージ最終部でワークエリアアドレスレジスタＲｘに、自己パッケージ使用ワークエリアサイズを加算することにより、次のパッケージで使用するワークエリアの先頭アドレスを与える。

これによって上記実施例に係わるプログラム作成装置および制御装置では、制御タスクプログラミング時のパッケージのワークエリアアドレス指定の負担がなくなり、加えてワークエリアもパッキング対象となるためにプログラムメモリだけでなく、ワークエリアのメモリをも有効に活用でき、プログラム作成・変更に柔軟に対応できる。

［発明の効果］

以上説明したように本発明によれば次のような種々の効果を奏する。

先ず、請求項１の発明においては、制御タスクのコンパイル時にタスクで使用する総ワークエリアサイズを演算し、パッケージのコンパイル時に、パッケージ最終部にワークエリアアドレス加算処理を付加し、制御装置にダウンローディングする際にはタスクサイズとワークエリアサイズとをタスクオブジェクトと共にダウンローディングするようにしているので、制御装置において制御タスク毎のワークエリアをワークエリアメモリ上に自動割付け可能となる。

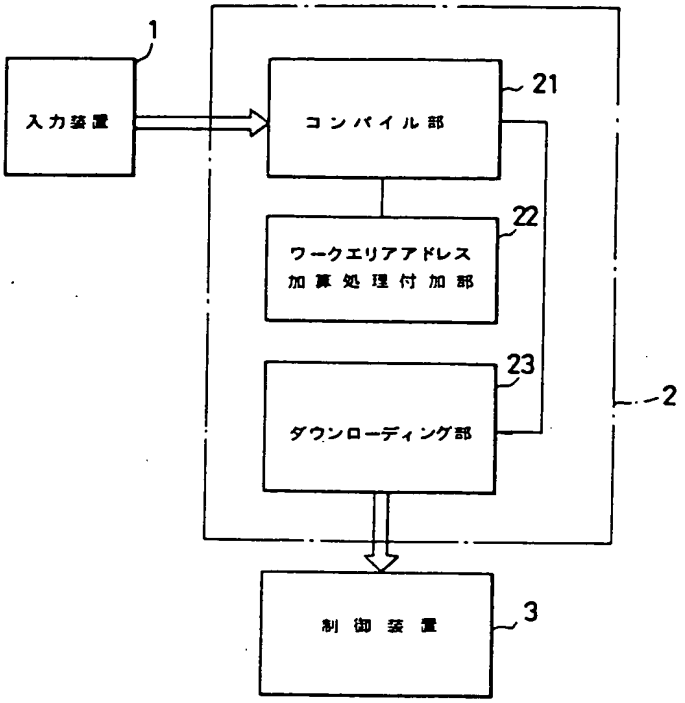
次に、請求項２，３の発明では、プログラム作成装置によって作成されたダウンローディングされてきた制御タスクに対し、そのダウンローディング時にプログラムメモリ、ワークエリアメモリの空きエリアへの自動割付けができ、プログラマがプログラム作成時に予めワークエリアを設定する必要がなく、ひいては設定ミスを撲滅できるとともに、メモリの有効活用が図れ、さらにプログラマの負担を軽減できてエンジニアリングコストを低減化できる。

４．図面の簡単な説明

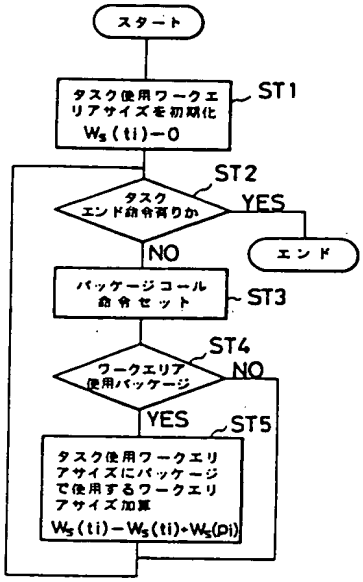
第１図は請求項１の発明に係わるプログラム作成装置の一実施例を示すブロック図、第２図は請求項２の発明に係わる制御装置の一実施例を示すブロック図、第３図はプログラム作成装置の制御タスクのコンパイル動作を説明するフロチャート、第４図はプログラム作成装置のパッケージのコンパイル動作を説明するフロチャート、第５図は制御装置のプログラム実行動作を説明する図、第６図はプログラム作成装置と制御装置とを統合した系別処理機能分担を示す説明図である。

１…入力装置、２…プログラム作成装置、３…制御装置、４…制御対象、５…タスクオブジェクト、６…パッケージオブジェクト、２１…コンパイル部、２２…ワークエリアアドレス加算処理付加部、２３…ダウンローディング部、３１…入力部、３２…プログラムメモリ、３３…プログラムメモリアドレス割付け部、３４…ワークエリアメモリ、３５…ワークエリアメモリ割付け部、３６…パッキング部、３７…プログラムメモリアドレス再割付け部、３８…ワークエリアメモリ再割付け部、３９…管理テーブルメモリ、４０…管理テーブル作成部、４１…タスクスケスケジューラ、４２…ワークエリアアドレスレジスタ。

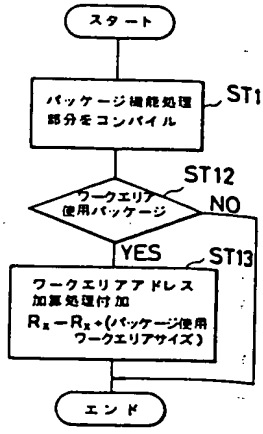
出願人代理人　弁理士　鈴江武彦



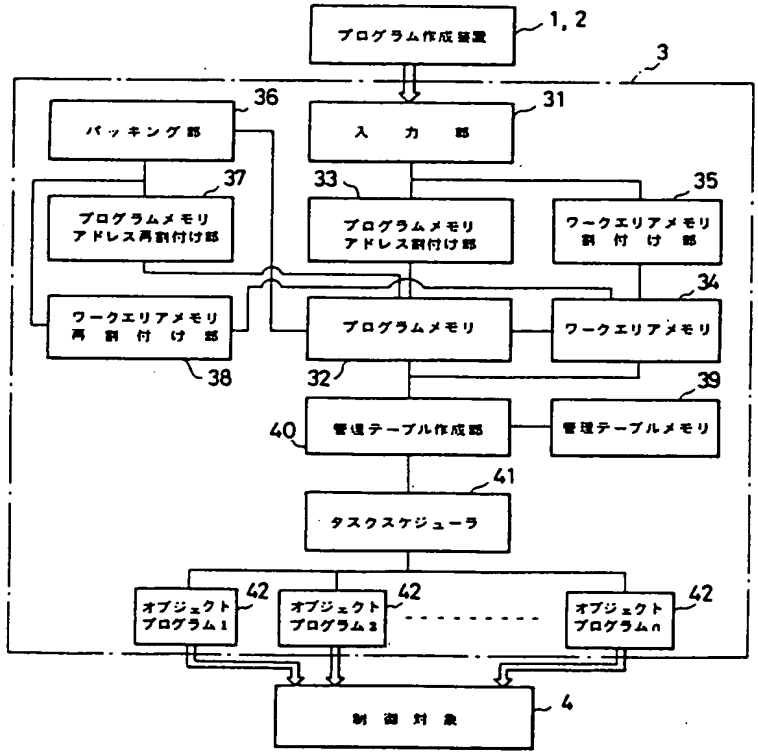
第 1 図



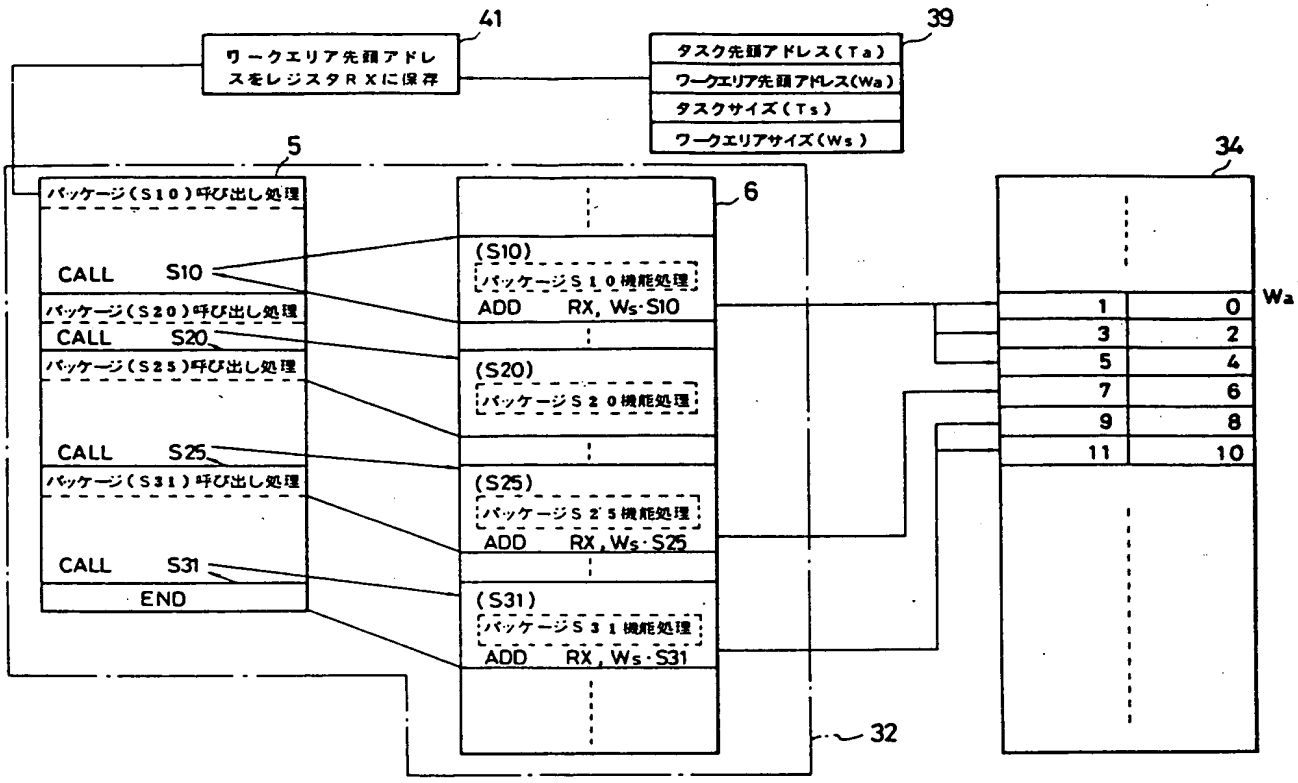
第 3 図



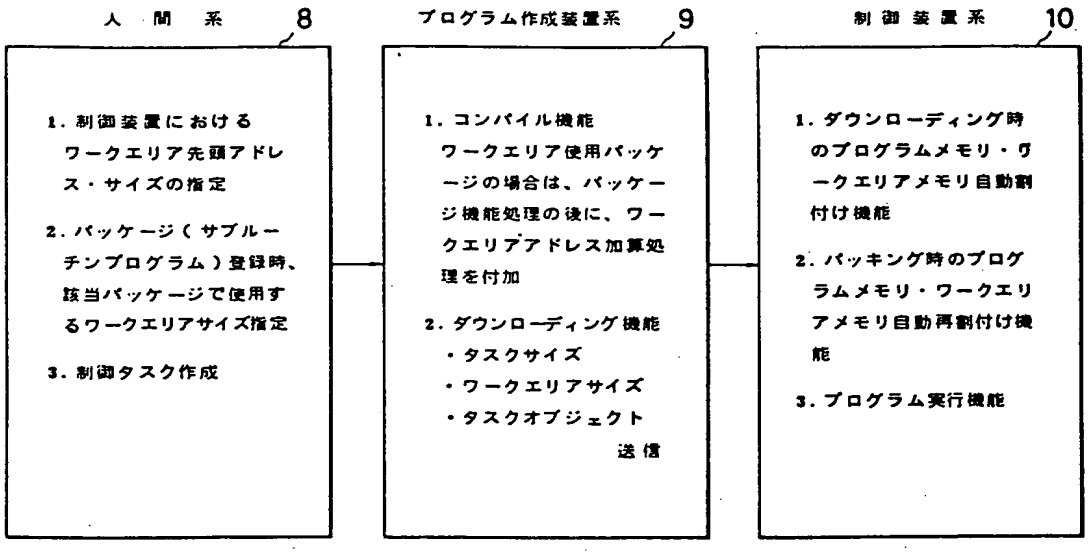
第 4 図



第 2 図


41
ワークエリア先頭アドレスをレジスタRXに保存
39
タスク先頭アドレス(Ta)
ワークエリア先頭アドレス(Wa)
タスクサイズ(Ts)
ワークエリアサイズ(Ws)
5
パッケージ(S10)呼び出し処理
CALL S10
パッケージ(S20)呼び出し処理
CALL S20
パッケージ(S25)呼び出し処理
CALL S25
パッケージ(S31)呼び出し処理
CALL S31
END
6
(S10)
パッケージS10機能処理
ADD RX, Ws·S10
(S20)
パッケージS20機能処理
(S25)
パッケージS25機能処理
ADD RX, Ws·S25
(S31)
パッケージS31機能処理
ADD RX, Ws·S31
32
34
Wa
1 0
3 2
5 4
7 6
9 8
11 10

第 5 図


人 間 系
8
1. 制御装置におけるワークエリア先頭アドレス・サイズの指定
2. パッケージ(サブルーチンプログラム)登録時、該当パッケージで使用するワークエリアサイズ指定
3. 制御タスク作成
プログラム作成装置系
9
1. コンパイル機能
ワークエリア使用パッケージの場合は、パッケージ機能処理の後に、ワークエリアアドレス加算処理を付加
2. ダウンローディング機能
・タスクサイズ
・ワークエリアサイズ
・タスクオブジェクト
送信
制御装置系
10
1. ダウンローディング時のプログラムメモリ・ワークエリアメモリ自動割付け機能
2. パッキング時のプログラムメモリ・ワークエリアメモリ自動再割付け機能
3. プログラム実行機能

第 6 図